arXiv:1209.0414v1 [math.CT] 3 Sep 2012

A direct proof that the category of 3-computads is not cartesian closed

Eugenia Cheng
Department of Mathematics, University of Sheffield
E-mail: e.cheng@sheffield.ac.uk

September 4, 2012

Abstract

We prove by counterexample that the category of 3-computads is not cartesian closed, a result originally proved by Makkai and Zawadowski. We give a 3-computad B and show that the functor $\_ \times B$ does not have a right adjoint, by giving a coequaliser that is not preserved by it.

Contents

Introduction

Makkai and Zawadowski proved in [7] that the category of (strict) 3-computads is not cartesian closed and hence is not a presheaf category. The result can be considered surprising—for example, the opposite was erroneously claimed in [3] (and corrected after Makkai and Zawadowski, in [4]).

The reason is related to the Eckmann-Hilton argument, but the proof given in [7], while having this reason at its heart, uses some sophisticated technology to bring this "reason" to fruition—some technical results of [3] for Artin glueing, which in turn rely on some technical results of Day [6].

In this paper we give a direct counterexample, that is, we give a 3-computad B and a coequaliser

$$E \rightrightarrows A \longrightarrow C$$

that is not preserved by the functor $\_ \times B$, hence $\_ \times B$ does not have a right adjoint.

The idea behind this counterexample is the same as the idea behind the proof in [7], and the result is, evidently, not new. However, we believe it is of value to provide this direct argument.

The root of the problem is that 2-cells having 1-cell identities as source and target do not behave "geometrically"—by an Eckmann-Hilton argument, horizontal and vertical composition for such cells must be the same and commutative. Intuitively, this means that cells do not have well-defined "shape"; a little more precisely, this means for example that if we have 2-cells a and b with identity source and target, then a 3-cell with source ab $(= ba)$ cannot have well-defined faces, as we cannot put the putative faces a and b in any order.

This argument obviously does not constitute a proof, but it is the idea at the root of the argument in [7] and at the root of the argument we give here. We begin in Section 1 by recalling the basic definitions; in Section 2 we give the counterexample, and in Section 3 we give the justification. Experts will only need to read Section 2.

Note that unless otherwise stated, all n-categories are strict.

Acknowledgements

I would like to thank and François Métayer for asking me for this counterexample. I would also like to thank Albert Burroni and Yves Guiraud for lively discussions around the subject.

1 Basic definitions

We begin by recalling the definition of the category of 3-computads. However, we will only need a small fragment of it for our counterexample, so we will focus on that part. 2-computads are defined by Street in [8]; the higher-dimensional generalisation is given by Burroni under the name "polygraphs" in [2] (see also [1]).

The idea is that a 3-computad is a 3-category that is "level-wise free". From another point of view it is the underlying data for a 3-category in which k-cells are allowed to have source and target that are pasting diagrams of $(k-1)$-cells, rather than the single $(k-1)$-cells that are the only allowed source and target for globular sets. Crucially for us, this means in particular that the source and target can be degenerate, that is, identities.

The definition proceeds inductively. At each dimension we must specify the k-cells and then generate pasting diagrams freely in order to specify the boundaries of cells at the next dimension. This is done using a free 3-category functor and is the technically tricky part of the definition. However, we will not actually need the full construction of this functor.

Definition 1.1. A **3-computad** A is given by, for each $0 \leq k \leq 3$

- a set A_k of k-cells, and
- a boundary map $A_k \longrightarrow PA_{k-1}$.

Here PA_{k-1} denotes the set of parallel pairs of formal composites of $(k-1)$-cells of A. A **morphism of 3-computads** $A \longrightarrow B$ is given by, for each $0 \leq k \leq 3$ a morphism

$$f_k : A_k \longrightarrow B_k$$

making the obvious squares commute. We write **3Comp** for the category of 3-computads and their morphisms.

In general it is quite complicated to make P precise, but each of the computads involved in our counterexample will have only one 0-cell and no 1-cells. In this case, the free 2-category on the 2-dimensional data is simply the free commutative monoid on A_2 (regarded as a doubly degenerate 2-category). We use the following terminology.

Definition 1.2. A 3-computad A is called **2-degenerate** if A_0 is terminal and A_1 is empty. Thus by the Eckmann-Hilton argument it consists of

- sets A_2 and A_3, equipped with
- source and target maps
$$A_3 \underset{t}{\overset{s}{\rightrightarrows}} A_2^*$$
where A_2^* denotes the free commutative monoid on A_2.

A morphism $A \longrightarrow B$ of such 3-computads is given by morphisms

$$A_2 \xrightarrow{f_2} B_2$$

$$A_3 \xrightarrow{f_3} B_3$$

such that the following diagram commutes serially.

$$\begin{tikzcd} A_3 \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f_3"'] & A_2^* \arrow[d, "f_2^*"] \\ B_3 \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & B_2^* \end{tikzcd}$$

2 The counterexample

All the 3-computads involved here will be 2-degenerate. When we check universal properties we will of course need to check them against all computads *a priori*, but we quickly see that the diagrams will ensure 2-degeneracy of any 3-computads involved.

We will write 2-cells as $a, b, \ldots$ and the commutative composition as

$$a.b = b.a.$$

In all that follows, every 3-cell will have a single 2-cell as target, but this is largely to ease the notation; a "smaller" counterexample would be possible with empty targets.

To show that **3Comp** is not cartesian closed we need to show that there exists $B \in \mathbf{3Comp}$ such that $\_ \times B$ does not have a right adjoint, so it suffices for $\_ \times B$ not to preserve all colimits. So we exhibit a coequaliser

$$\begin{tikzcd} E \arrow[r, shift left, "\alpha_1"] \arrow[r, shift right, "\alpha_2"'] & A \arrow[r, "\beta"] & C \end{tikzcd}$$

and a computad B such that the functor $\_ \times B$ does not preserve it.

Step 1: the coequaliser

1. Let A be the 2-degenerate 3-computad with 2-cells a_1, a_2, a_3 and a single 3-cell
$$a_1.a_2 \xrightarrow{f} a_3.$$
2. Let E be the 2-degenerate 3-computad with 2-cells x, y and no 3-cells.
3. Define the morphism α_1 by
$$\begin{array}{ccc} x & \longmapsto & a_1 \\ y & \longmapsto & a_3 \end{array}$$
and define α_2 by
$$\begin{array}{ccc} x & \longmapsto & a_2 \\ y & \longmapsto & a_3 \end{array}$$
4. Thus the coequaliser C simply identifies a_1 and a_2; it has 2-cells $\bar{a}, a_3$ and a single 3-cell
$$\bar{a}.\bar{a} \xrightarrow{\bar{f}} a_3.$$

Step 2: the functor $\_ \times B$

5. Let B be the 2-degenerate 3-computad (isomorphic to A) with 2-cells b_1, b_2, b_3 and a single 3-cell
$$b_1.b_2 \xrightarrow{g} b_3.$$
6. $E \times B$ has 2-cells (x, b_j) and (y, b_j) for $j = 1, 2, 3$. It has no 3-cells.

7. $A \times B$ is the key structure. It has 2-cells (a_i, b_j) for $i, j = 1, 2, 3$ and *two* 3-cells

$$\begin{array}{ccc} (a_1, b_1).(a_2, b_2) & \xrightarrow{(f,g)_1} & (a_3, b_3) \\ (a_2, b_1).(a_1, b_2) & \xrightarrow{(f,g)_2} & (a_3, b_3) \end{array}$$

This is probably the most interesting part of the argument; we give the full proof later.

8. $C \times B$ has 2-cells $(\bar{a}, b_j)$ for $j = 1, 2, 3$ and a single 3-cells

$$(\bar{a}, b_1).(\bar{a}, b_2) \xrightarrow{(\bar{f},g)} (a_3, b_3).$$

Step 3: non-preservation

9. We now examine the coequaliser

$$E \times B \underset{\alpha_2 \times 1}{\overset{\alpha_1 \times 1}{\rightrightarrows}} A \times B \longrightarrow P$$

and show that it is not isomorphic to $C \times B$.

Now the morphism $\alpha_1 \times 1$ is given by

$$\begin{array}{ccc} (x, b_j) & \longmapsto & (a_1, b_j) \\ (y, b_j) & \longmapsto & (a_3, b_j) \end{array}$$

and $\alpha_2 \times 1$ by

$$\begin{array}{ccc} (x, b_j) & \longmapsto & (a_2, b_j) \\ (y, b_j) & \longmapsto & (a_3, b_j) \end{array}$$

Thus the coequaliser P simply identifies (a_1, b_j) with (a_2, b_j) for each j. So it has 2-cells which we may call $(\bar{a}, b_j)$ and (a_3, b_j) (which is to be expected as the coequaliser is preserved up to 2 dimensions).

P has two distinct 3-cells

$$\begin{array}{ccc} (\bar{a}, b_1).(\bar{a}, b_2) & \xrightarrow{(f,g)_1} & (a_3, b_3) \\ (\bar{a}, b_1).(\bar{a}, b_2) & \xrightarrow{(f,g)_2} & (a_3, b_3). \end{array}$$

Since $C \times B$ has only one 3-cell it is clear that $C \times B$ is not isomorphic to this coequaliser P, that is, $\_ \times B$ does not preserve the original coequaliser.

Note that the canonical factorisation

$$P \longrightarrow C \times B$$

identifies the 3-cells $(f, g)_1$ and $(f, g)_2$.

3 Universal properties

In this section we check all the universal properties required for the counterexample. In principle we only need to check the 3-cells, as 2-computads form a presheaf category so we know that the lower dimensions behave pointwise. However we include the full argument for completeness, and because it is straightforward.

Lemma 3.1. *The product $A \times B$ is as given in the previous section, with the obvious projections.*

Proof. We exhibit its universal property. Consider a 3-computad Y and morphisms

$$\begin{tikzcd} & Y \arrow[d, dashed, "k"] \arrow[ddl, bend right, "u"'] \arrow[ddr, bend left, "v"] & \\ & A \times B \arrow[dl, "p"'] \arrow[dr, "q"] & \\ A & & B \end{tikzcd}$$

We seek to exhibit a unique factorisation k as shown. On 0-, 1- and 2-cells, $A \times B$ is just a product, so we define the factorisation at these dimensions as for products ie

$$k(t) = (u(t), v(t)).$$

Note in particular that A and B have no 1-cells, so for the morphisms u and/or v to exist, Y cannot have any 1-cells either. So this map respects boundaries trivially.

We now discuss the factorisation on 3-cells. Let e be a 3-cell in Y. Now A and B have only one 3-cell each, f and g respectively. So we must have

$$\begin{aligned} u(e) &= f \\ v(e) &= g \end{aligned}$$

thus e must have boundary as follows

$$y_1 \cdot y_2 \xrightarrow{e} y_3$$

for some 2-cells $y_1, y_2, y_3 \in Y$. Then since the action of u and v respect the boundary of e we know y_3 must be sent to a_3 and b_3 respectively. However considering the source there is some ambiguity as the product is commutative, so for each of u and v there are two possibilities—either the subscripts are left the same, or they are switched. That is, on ordered pairs the action of u is

$$\begin{aligned} \text{either} \quad (y_1, y_2) &\longmapsto (a_1, a_2) \\ \text{or} \quad (y_1, y_2) &\longmapsto (a_2, a_1) \end{aligned}$$

and similarly the action of v is

$$\begin{array}{rrcl} \text{either} & (y_1, y_2) & \longmapsto & (b_1, b_2) \\ \text{or} & (y_1, y_2) & \longmapsto & (b_2, b_1). \end{array}$$

There are thus 4 cases, but in each case $k(e)$ is uniquely determined to be either $(f,g)_1$ or $(f,g)_2$ by the condition that k preserves boundary. Explicitly, $k(e)$ is specified by examining the action of u and v as shown by the following table.

		v	
		$(y_1, y_2) \longmapsto (b_1, b_2)$	$(y_1, y_2) \longmapsto (b_2, b_1)$
u	$(y_1, y_2) \longmapsto (a_1, a_2)$	$(f,g)_1$	$(f,g)_2$
	$(y_1, y_2) \longmapsto (a_2, a_1)$	$(f,g)_2$	$(f,g)_1$

$\square$

The other products follow similarly, but more easily. It remains to check the universal properties of the two coequalisers in question, which is much more straightforward.

Consider a diagram

$$\begin{tikzcd} E \arrow[r, shift left, "\alpha_1"] \arrow[r, shift right, "\alpha_2"'] & A \arrow[r, "q"] \arrow[dr, "u"'] & C \arrow[d, dashed, "k"] \\ & & Y \end{tikzcd}$$

with $u\alpha_1 = u\alpha_2$. We seek a unique factorisation k as shown.

- On 0-cells: A and C only have one 0-cell each; writing each as $*$ we must have $k(*) = u(*) \in Y$.
- On 1-cells: A and C have no 1-cells, so as before Y cannot have any either.
- On 2-cells: To make the triangle commute we must put

  $$\begin{array}{rcl} k(\bar{a}) & = & u(a_1) \; [= u(a_2)] \\ k(a_3) & = & u(a_3). \end{array}$$

  This respects boundaries as all 2-cells involved are degenerate.
- On 3-cells: To make the triangle commute, we must have $k(\bar{f}) = u(f)$. This respects boundaries, by our definition of k on 2-cells.

The other coequaliser proceeds in the same way, but with two 3-cells.

Remark 3.2. Note that this sort of counterexample cannot arise for 2-computads, as 2 is the lowest dimension of cell for which the Eckmann-Hilton argument can be used. Note also that this problem does not arise for weak 3-computads as weak identity 1-cells impede the Eckmann-Hilton argument on degenerate 2-cells. This difference between the commutativity of degenerate 3-cells in weak and strict structures also arises in [5].

References

[1] M. A. Batanin. Computads for finitary monads on globular sets. *Contemporary Mathematics*, 210:37–57, 1998.

[2] A. Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, 1993.

[3] Aurelio Carboni and Peter Johnstone. Connected limits, familial representability and Artin glueing. *Mathematical Structures in Computer Science*, 5(4):441–459, 1995.

[4] Aurelio Carboni and Peter Johnstone. Corrigenda for "Connected limits, familial representability and Artin glueing". *Mathematical. Structures in Comp. Sci.*, 14(1):185–187, 2004.

[5] Eugenia Cheng and Michael Makkai. A note on Penon's definition of weak n-category. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 50:83–101, 2009.

[6] Brian Day. Enriched Tannaka reconstruction. *Journal of Pure and Applied Algebra*, 108:17–22, 1969.

[7] Michael Makkai and Marek Zawadowski. 3-computads do not form a presheaf category. *Journal of Pure and Applied Algebra*, 212(11):2543–3546, 2008.

[8] Ross Street. Limits indexed by category valued 2-functors. *Journal of Pure and Applied Algebra*, 8:149–181, 1976.